

Supplementary Information

Identifiable learning of dissipative dynamics

Aiqing Zhu¹, Beatrice W. Soh^{2,4}, Grigorios A. Pavliotis^{3,*}, and Qianxiao Li^{1,4,*}

¹Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, 119076, Singapore

²Department of Chemical and Biomolecular Engineering, National University of Singapore, 4 Engineering Drive 4, 117585, Singapore

³Department of Mathematics, Imperial College London, London, SW7 2AZ, United Kingdom

⁴Institute for Functional Intelligent Materials, National University of Singapore, 4 Science Drive 2, 117544, Singapore

** Corresponding authors: g.pavliotis@imperial.ac.uk, qianxiao@nus.edu.sg*

Contents

A Expressivity of Network Modules	2
A.1 Expressivity of σ_θ	2
A.2 Expressivity of V_θ	2
B Hyperparameter	4
C Pseudocode	4
D Supplementary Information for Polymer Stretching Dynamics	4
E Supplementary Information for Stochastic Gradient Langevin Dynamics	6
F Supplementary Information for Synthetic Examples	11
F.1 Linear Case	11
F.2 Nonlinear Case	12
F.3 A twenty-dimensional nonlinear SDE	17

A Expressivity of Network Modules

A.1 Expressivity of σ_θ

The parameterization of σ_θ is based on a Cholesky-type representation. For any symmetric positive-definite matrix $M(Z)$, the matrix $2M(Z)$ has a unique Cholesky factorization

$$2M(Z) = L(Z)L(Z)^\top,$$

where $L(Z)$ is lower triangular with strictly positive diagonal entries. In our construction, the network output σ_1 parameterizes the strictly lower-triangular entries of $L(Z)$, whereas the diagonal entries are represented by applying the smooth positive map

$$\rho(x) = \frac{\sqrt{x^2 + 1} + x}{2}$$

to the network output σ_2 . Since $\rho : \mathbb{R} \rightarrow (0, \infty)$, this positivity constraint does not reduce expressivity for strictly positive diagonal factors. Consequently, with sufficiently expressive networks, σ_θ can approximate the Cholesky factor of any state-dependent symmetric positive-definite matrix. Equivalently, the induced matrix $M_\theta = \frac{1}{2}\sigma_\theta\sigma_\theta^\top$ can approximate the target diffusion matrix itself.

A.2 Expressivity of V_θ

The expressivity of this parameterization should be understood in a local, up-to-constant sense. Let $K \subset \mathbb{R}^D$ be compact and let $V^* \in C(K)$ be a target potential. Since additive constants do not affect either the drift generated by ∇V^* or the corresponding normalized density, choose a constant c_K such that

$$G(Z) := V^*(Z) + c_K - \beta_3 \|Z\|^2 \geq 0, \quad Z \in K.$$

If the network class for U is a universal approximator on compact sets, we may choose one component of U , say U_1 , to approximate $\sqrt{2G/\beta_1} - \beta_2 \langle \gamma_1, Z \rangle$ uniformly on K , while fixing γ_1 and setting

the remaining components to zero. It follows that

$$\left\| \frac{\beta_1}{2} (U_1(Z) + \beta_2 \langle \gamma_1, Z \rangle)^2 - G(Z) \right\|_{C^1(K)} < \varepsilon$$

and therefore

$$\|V_\theta(Z) - (V^*(Z) + c_K)\|_{C^1(K)} < \varepsilon, \quad \text{and} \quad \|\nabla V_\theta(Z) - \nabla V^*(Z)\|_{C^0(K)} < \varepsilon.$$

This local approximation of the potential also yields a global approximation of the associated density. Let

$$p_\theta(Z) := \frac{e^{-V_\theta(Z)}}{\int_{\mathbb{R}^D} e^{-V_\theta(Y)} dY}, \quad p^*(Z) := \frac{e^{-V^*(Z)}}{\int_{\mathbb{R}^D} e^{-V^*(Y)} dY}.$$

For $Z \in K$, assume that $V_\theta(Z) = V^*(Z) + c_K + \delta(Z)$. Set

$$a := e^{\|\delta\|_{L^\infty(K)}} - 1, \quad \text{and} \quad b := \frac{e^{c_K} \int_{\mathbb{R}^D \setminus K} e^{-\beta_3 \|Y\|^2} dY + \int_{\mathbb{R}^D \setminus K} e^{-V^*(Y)} dY}{\int_{\mathbb{R}^D} e^{-V^*(Y)} dY}.$$

Then we have

$$\left| \frac{e^{c_K} \int_{\mathbb{R}^D} e^{-V_\theta(Y)} dY}{\int_{\mathbb{R}^D} e^{-V^*(Y)} dY} - 1 \right| \leq a + b,$$

where the error on $\mathbb{R}^D \setminus K$ is bounded by the two tail terms appearing in the definition of b ; and the error on K is bounded by

$$\int_K e^{-V^*(Y)} |e^{-\delta(Y)} - 1| dY \leq a \int_{\mathbb{R}^D} e^{-V^*(Y)} dY.$$

Given any $\varepsilon > 0$, choose K sufficiently large such that $b \leq \min\left\{\frac{1}{4}, \frac{\varepsilon}{8}\right\}$, and then choose the approximation on K sufficiently accurate such that

$$\|\delta\|_{L^\infty(K)} \leq \log\left(1 + \min\left\{\frac{1}{4}, \frac{\varepsilon}{8}\right\}\right), \quad \text{and thus } a = e^{\|\delta\|_{L^\infty(K)}} - 1 \leq \min\left\{\frac{1}{4}, \frac{\varepsilon}{8}\right\}.$$

Then $a + b \leq 1/2$. Denote $R := e^{c_K} \int_{\mathbb{R}^D} e^{-V_\theta(Y)} dY / \int_{\mathbb{R}^D} e^{-V^*(Y)} dY$, we obtain that

$$\int_K |p_\theta(Z) - p^*(Z)| dZ = \int_K p^*(Z) \left| \frac{e^{-\delta(Z)}}{R} - 1 \right| dZ \leq \frac{2a + b}{1 - a - b} \int_K p^*(Z) dZ \leq 4a + 2b,$$

where we have used the estimate that $|R - 1| \leq a + b$. On the tail region,

$$\int_{\mathbb{R}^D \setminus K} |p_\theta(Z) - p^*(Z)| dZ \leq \int_{\mathbb{R}^D \setminus K} p_\theta(Z) dZ + \int_{\mathbb{R}^D \setminus K} p^*(Z) dZ \leq 2b.$$

Therefore, we conclude that

$$\|p_\theta - p^*\|_{L^1(\mathbb{R}^D)} \leq 4a + 4b \leq \varepsilon.$$

Finally, It is remarked that the use of a sum over m squared components further enriches the class of non-negative functions that can be represented at finite width. The linear terms $\beta_2 \langle \gamma_i, Z \rangle$ provide an additional learnable quadratic channel at large $\|Z\|$. Their contribution includes the non-negative quadratic form $\frac{\beta_1 \beta_2^2}{2} \sum_{i=1}^m \langle \gamma_i, Z \rangle^2$, which strengthens the prescribed quadratic confinement beyond the isotropic baseline $\beta_3 \|Z\|^2$.

	Polymer Stretching Dynamics	SGLD-LSR	SGLD-ICA
<u>Potential network V_θ</u>			
Hidden widths of U	(128)	(128)	(128)
Activation	shifted ReQU	tanh	tanh
Embedding dimension m	32	32	32
β_1	$2^{\varepsilon/3}/3$	$1/\max\{b, 50\}$	1
β_2	1	$\max\{b, 50\}$	1
β_3	0.01	0.01	0.01
<u>Antisymmetric network H</u>			
Hidden widths	(128, 128)	(128, 128)	(128, 128)
Activation	tanh	tanh	tanh
<u>Diffusion σ_θ</u>			
Form	state-indep. diagonal	state-dep. lower-triangular	state-dep. lower-triangular
Hidden widths	—	(32, 32) each	(32, 32) each
Activation	—	tanh	tanh

Supplementary Table 1: Network hyperparameters used in the polymer stretching dynamics and SGLD experiments. The symbol “—” indicates that the corresponding component or parameter is not used.

	Polymer Stretching Dynamics	SGLD-LSR	SGLD-ICA
Optimizer	Adam	Adam	Adam
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
Batch size	2 trajectories	1×10^5	1×10^5
Number of epochs	2,000	500	500

Supplementary Table 2: Training hyperparameters used in the numerical experiments.

B Hyperparameter

The hyperparameter choices are summarized in Supplementary Table 1 and Supplementary Table 2.

C Pseudocode

To make the implementation procedure explicit, we present the pseudocode of the proposed UIDD workflow in Algorithm 1.

D Supplementary Information for Polymer Stretching Dynamics

Training is performed for 2000 epochs with a batch size of 2. Each batch contains two complete trajectories, which allows efficient optimization while preserving the full temporal structure of the training sequences.

To assess the accuracy of the learned UIDD, we evaluate its performance on both the training and test trajectories. Moreover, validation is performed using two distinct initial chain conformations absent from the training set. These representative configurations were deliberately initialized with near-identical projected extensions yet diverged significantly in their transient stretching dynamics, with one stretching much faster than the other. As shown in Supplementary Figure 2, on both the training and test datasets across all cases, UIDD precisely captures these contrasting behaviors and reproduces the true ensemble evolution of chain extension under varying external force. UIDD also achieves high-fidelity reconstruction of the first-passage time distributions required to attain a specified extension threshold.

Algorithm 1 UIDD training algorithm

Require: Discrete trajectories $\{z_i(t_k)\}$; observation step Δt ; model dimensions; neural-network architectures; optimizer settings; evaluation initial conditions.

Model parameterization:

- 1: Parameterize the potential $V_\theta(z)$, Hamiltonian functions $H_\theta(z) = (H_{\theta,1}(z), \dots, H_{\theta,D-1}(z))$, and diffusion factor $\sigma_\theta(z)$.
- 2: Construct

$$M_\theta(z) = \frac{1}{2} \sigma_\theta(z) \sigma_\theta(z)^\top, \quad W_\theta(z) = \sum_{d=1}^{D-1} H_{\theta,d}(z) J_d,$$

where each fixed antisymmetric matrix J_d couples coordinates d and $d+1$.

- 3: Define the model drift

$$f_\theta(z) = -[M_\theta(z) + W_\theta(z)] \nabla V_\theta(z) + \nabla \cdot M_\theta(z) + \nabla \cdot W_\theta(z).$$

Training:

- 4: Form adjacent one-step pairs $(z_i(t_k), z_i(t_{k+1}))$ from the trajectory data.
- 5: **for** each training epoch **do**
- 6: **for** each mini-batch $\mathcal{B}$ of one-step pairs **do**
- 7: Minimize the Euler–Maruyama negative log-likelihood

$$\mathcal{L}(\theta) = \sum_{(z, z') \in \mathcal{B}} \left[\frac{1}{2} \log \det(2M_\theta(z) \Delta t) + \frac{1}{2} \|z' - z - f_\theta(z) \Delta t\|_{(2M_\theta(z) \Delta t)^{-1}}^2 \right],$$

- 8: **end for**

- 9: **end for**

Analysis

- 10: Stationary density

$$\rho_\theta(z) \propto e^{-V_\theta(z)}.$$

- 11: Reversible and irreversible drift components

$$f_{\text{rev},\theta} = -M_\theta \nabla V_\theta + \nabla \cdot M_\theta, \quad f_{\text{irr},\theta} = -W_\theta \nabla V_\theta + \nabla \cdot W_\theta.$$

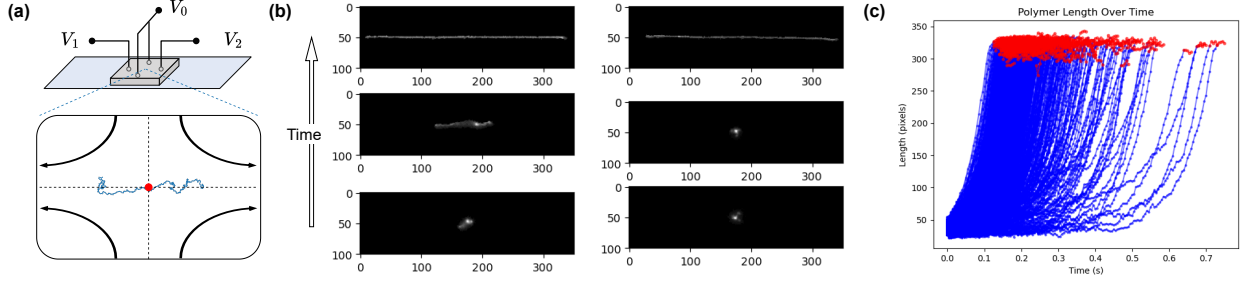
- 12: Estimate the global EPR by Monte Carlo integration of

$$\dot{S}_{\text{tot}} = \int f_{\text{irr},\theta}(z)^\top M_\theta(z)^{-1} f_{\text{irr},\theta}(z) \rho_\theta(z) dz.$$

To further evaluate accuracy of the the learned SDE, we compare the empirical distributions of the model predictions and the reference ensemble at each observation time using the Gaussian-kernel empirical squared maximum mean discrepancy. The simulated trajectories are evaluated on the same time grid as the data. The evaluation is repeated over four training seeds, and we report the mean together with a one-standard-deviation band.

At each observation time t_k , the metric is defined as

$$\text{MMD}^2(t_k) = \frac{1}{N^2} \sum_{i,j=1}^N k(x_i(t_k), x_j(t_k)) + \frac{1}{N^2} \sum_{i,j=1}^N k(\hat{x}_i(t_k), \hat{x}_j(t_k)) - \frac{2}{N^2} \sum_{i,j=1}^N k(x_i(t_k), \hat{x}_j(t_k)),$$



Supplementary Figure 1: **(a)** The schematic and photograph show our experimental system. A cross-slot microfluidic device is equipped with platinum electrodes in each of its four reservoirs. We create an electric field by applying positive voltages V_1 and V_2 to the east (E) and west (W) reservoirs, while the north (N) and south (S) reservoirs serve as the ground. This voltage configuration guides negatively charged DNA from the grounded reservoirs, pulling it into the channel's center. The process culminates when a single molecule is captured and held in place, a point marked by the blue dot in the schematic. **(b)** Snapshots showing two distinct DNA stretching processes. **(c)** Time evolution of DNA length, The red points correspond to the selected DNA configurations used to evaluate the global EPR.

where

$$k(u, v) = \exp\left(-\frac{\|u - v\|^2}{\sigma^2}\right),$$

and $\{x_i(t_k)\}_{i=1}^N$ denotes the reference trajectories generated by the target SDE, whereas $\{\hat{x}_i(t_k)\}_{i=1}^N$ denotes the corresponding predictions generated by the learned model. A single bandwidth, shared across all time points, is fixed by the median heuristic on a pooled set of up to 4×10^3 states sampled from an independent target ensemble.

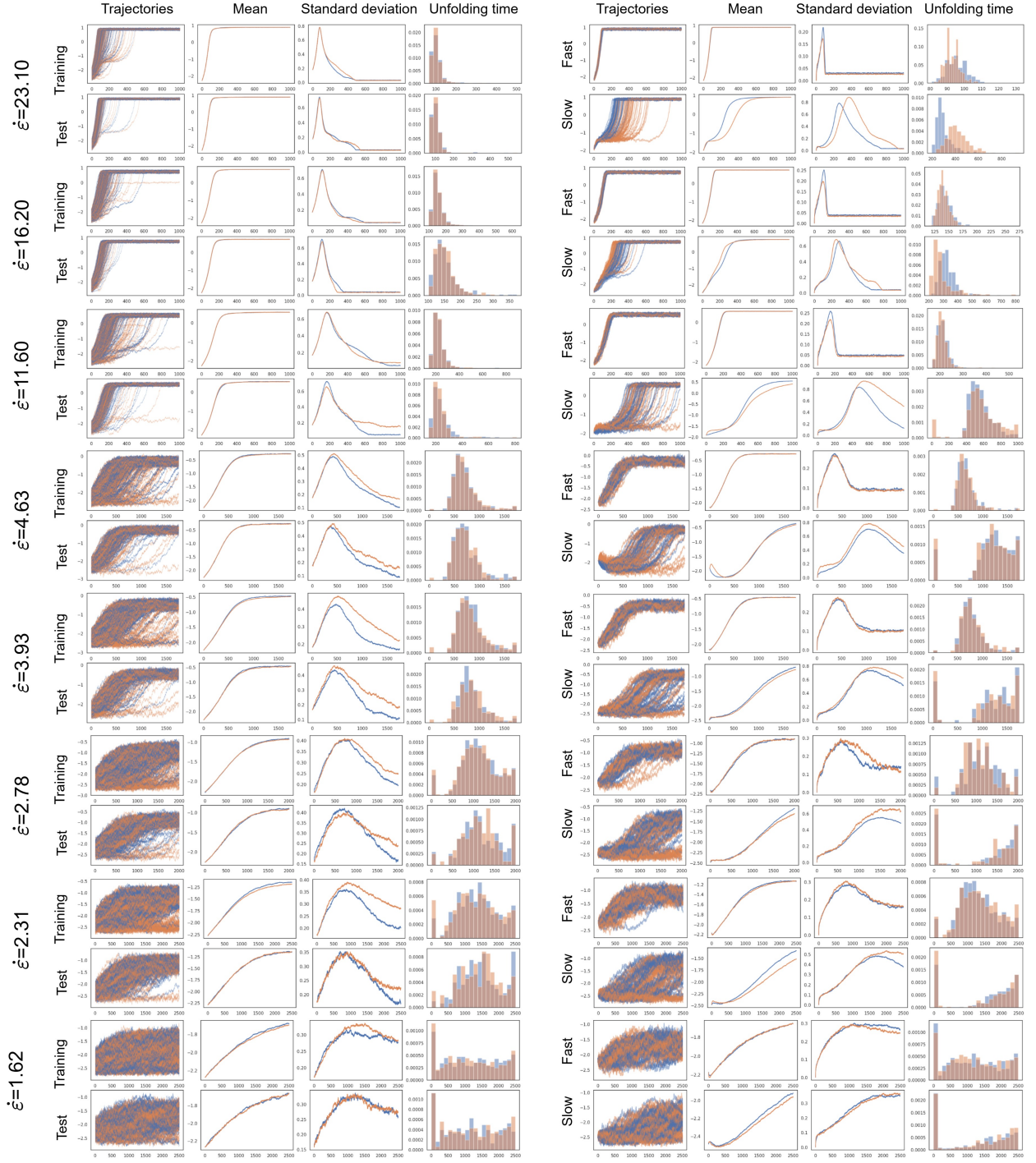
Supplementary Figure 4 reports the resulting MMD² curves. Across all strain rates, the curves rise at early times and then approach a plateau. The plateau level depends mainly on the strain rate and is larger at lower strain rates, where the reference distribution is broader. Overall, UIDD, OnsagerNet, and SDENet show comparable distribution-matching performance: neither model consistently outperforms the others, and their seed-to-seed variability bands largely overlap. Thus, the identifiable structure in UIDD does not systematically improve or degrade distribution matching relative to the OnsagerNet and SDENet baselines.

For completeness, we provide the results for all examined strain rates $\dot{\epsilon}$ in Supplementary Figure 3. The behavior of the total EPR remains qualitatively consistent with that reported in the main text, exhibiting a transition from nearly reversible dynamics at small $\dot{\epsilon}$ to strongly irreversible unfolding at large $\dot{\epsilon}$.

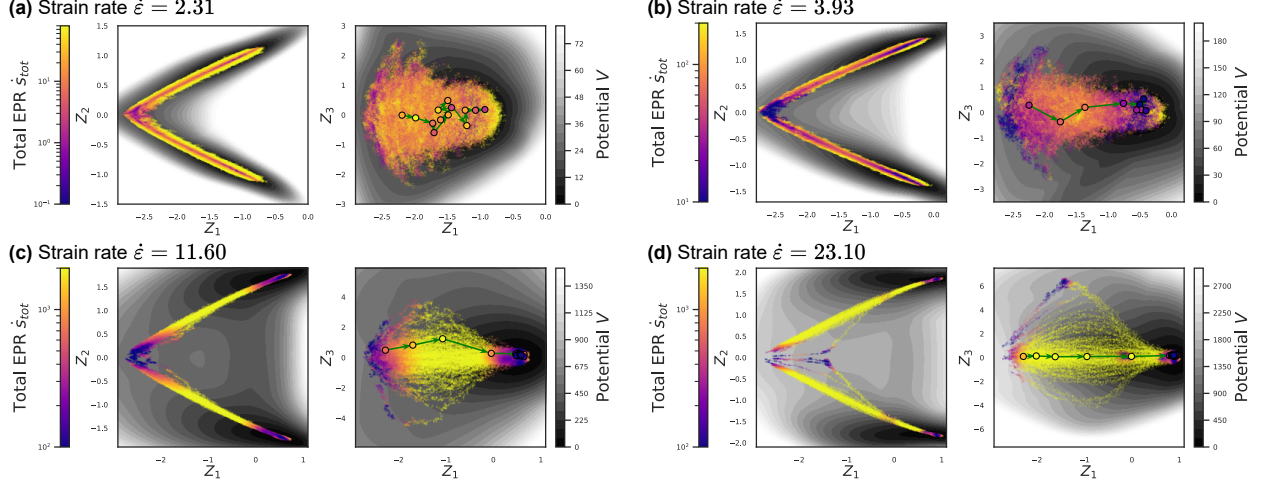
E Supplementary Information for Stochastic Gradient Langevin Dynamics

Here the model is trained for 500 epochs using the Adam optimizer with a batch size of 100,000. Each batch consists of independently sampled paired data points.

To evaluate the robustness and reconstruction fidelity of UIDD in the context of least-squares regression with SGLD, we examine its ability to reproduce both transient trajectories and output



Supplementary Figure 2: Comparison of predicted and true stretching dynamics, together with their statistical characteristics, for training trajectories, standard test trajectories, and supplementary fast and slow test trajectories.



Supplementary Figure 3: The total EPR $\dot{s}_{\text{tot}}$ on test trajectories projected onto the Z_1 - Z_2 and Z_1 - Z_3 planes. The background is the corresponding potential energy surface. One example trajectory from the test dataset is overlaid on the Z_1 - Z_3 panels.

losses across diverse parameter settings (Supplementary Figure 5). UIDD consistently recovers the temporal evolution of the stochastic dynamics under different batch sizes and initialization schemes, demonstrating that the learned representation captures the underlying generative mechanism rather than overfitting to specific training conditions. This robustness enables a mechanistic interpretation of SGLD behavior directly from the learned model.

We further validate UIDD on the nonlinear ICA benchmark (Supplementary Figure 6). Across varying batch sizes and initialization schemes, the model yields highly consistent reconstructions of both the transient trajectories and the invariant distributions. These results confirm that UIDD accurately captures the stochastic dynamics of the ICA with SGLD.

Theoretical Justification for EPR as a Measure of Sampling Error For convenience, we repeat the dynamics of UIDD:

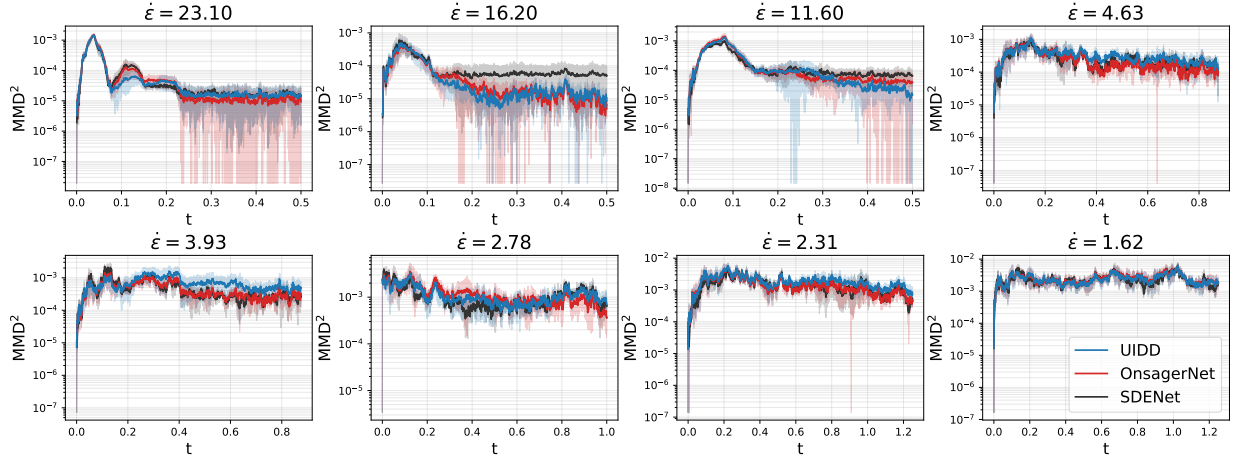
$$\dot{\mathbf{Z}}_t = -[M(\mathbf{Z}_t) + W(\mathbf{Z}_t)]\nabla V(\mathbf{Z}_t) + \nabla \cdot M(\mathbf{Z}_t) + \nabla \cdot W(\mathbf{Z}_t) + \sigma(\mathbf{Z}_t)\dot{\mathbf{B}}_t. \quad (1)$$

We also recall that the discrete-time dynamics of SGLD converge weakly, in the vanishing step-size limit, to a continuous-time diffusion process governed by the Itô SDE [1]:

$$\dot{\mathbf{Z}} = -\nabla L(\mathbf{Z}) + (\eta \Sigma(\mathbf{Z}) + 2)^{\frac{1}{2}} \dot{\mathbf{B}}_t, \text{ with } \Sigma(\mathbf{Z}) = \text{Cov}\left[\frac{n}{b} \sum_{i=1}^b \nabla l_{\gamma_i}(\mathbf{Z})\right],$$

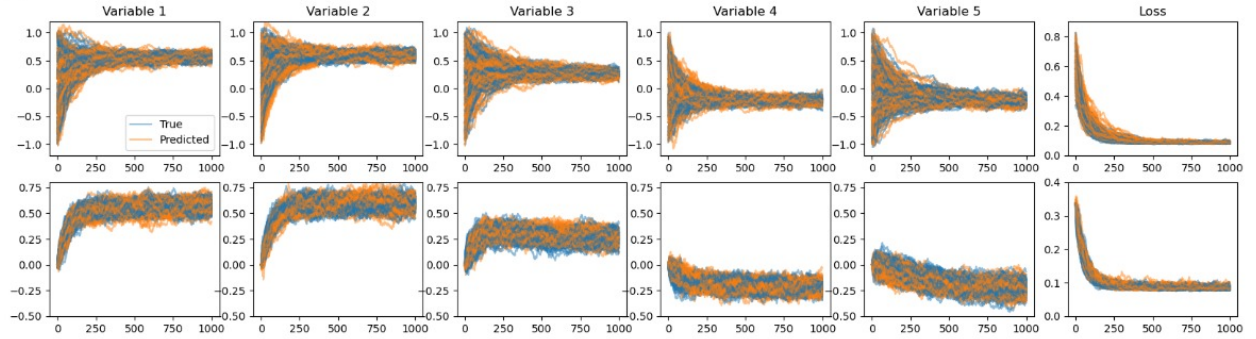
where $\{\gamma_i\}_{i=1}^b$ indexes a mini-batch of size b . By Theorem 1, there exists a matrix field $W_b(\mathbf{Z})$ of the form

$$W(\mathbf{Z}) = \begin{pmatrix} 0 & H_1(\mathbf{Z}) & \cdots & 0 & 0 \\ -H_1(\mathbf{Z}) & 0 & \cdots & 0 & 0 \\ 0 & -H_2(\mathbf{Z}) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & H_{D-1}(\mathbf{Z}) \\ 0 & 0 & \cdots & -H_{D-1}(\mathbf{Z}) & 0 \end{pmatrix}.$$

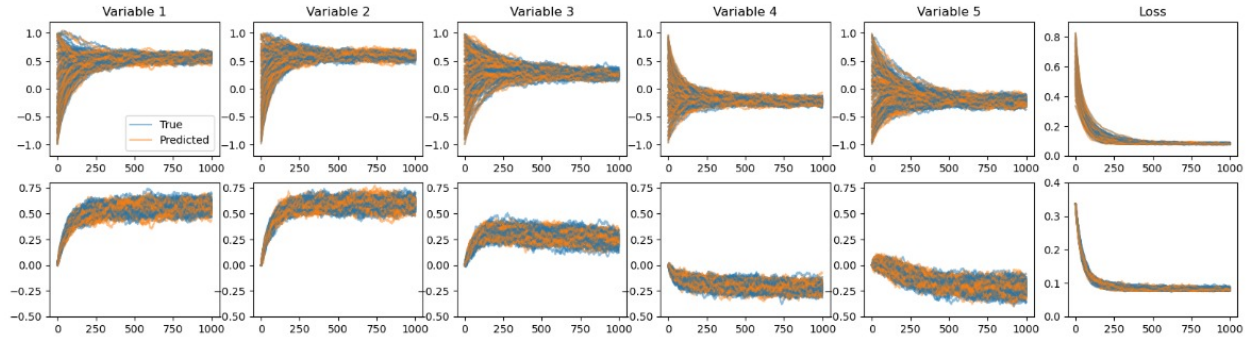


Supplementary Figure 4: Prediction accuracy for the polymer stretching dynamics, measured by Gaussian-kernel MMD^2 over time. Curves show the mean over four training seeds, with shaded regions indicating one standard deviation.

(a) Batch size: 1

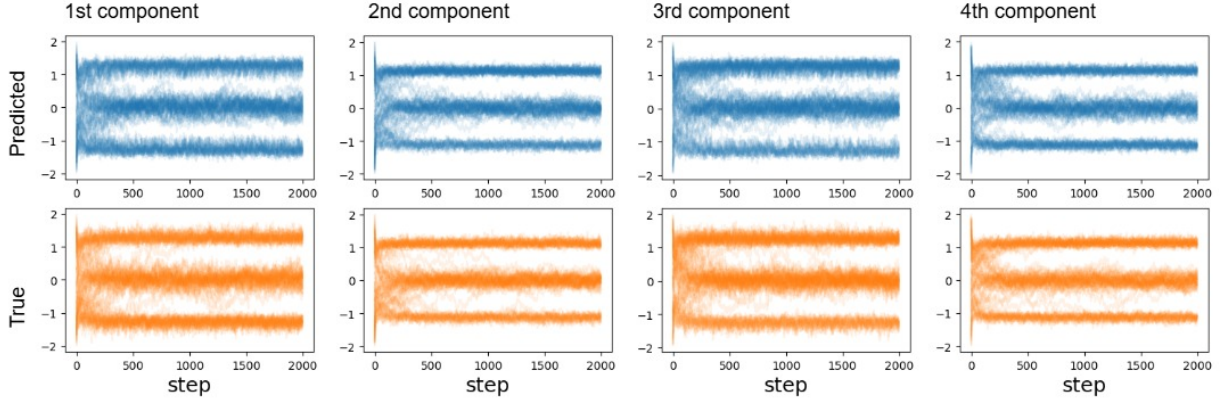


(b) Batch size: 64

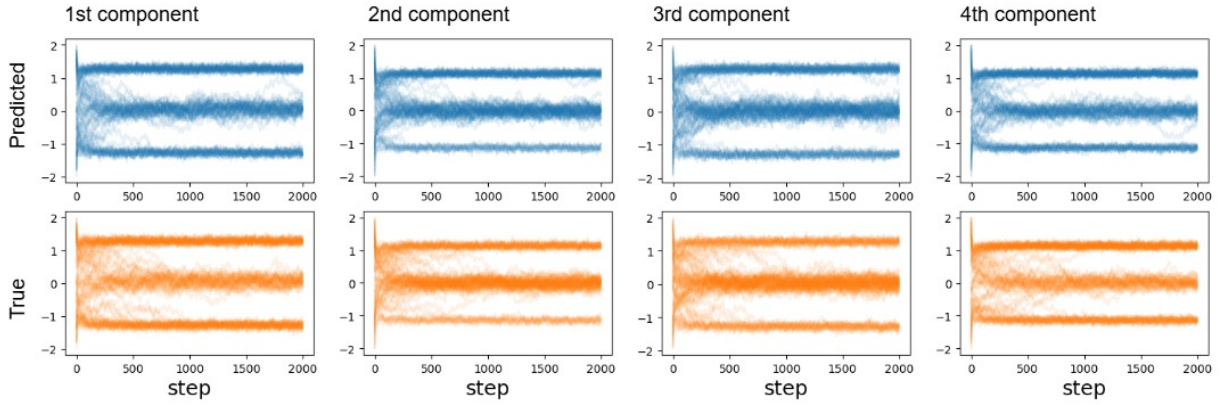


Supplementary Figure 5: Predicted versus true least squares regression with SGLD trajectories under varying batch sizes and initialization. All trajectories are linearly rescaled to $[-1, 1]$ and averaged over 500 independent runs. Panels (a) show results for batch size 1, with either random initialization from $[-1, 1]$ (top) or all coordinates set to zero (bottom). Panels (b) correspond to batch size 64, presented in the same order.

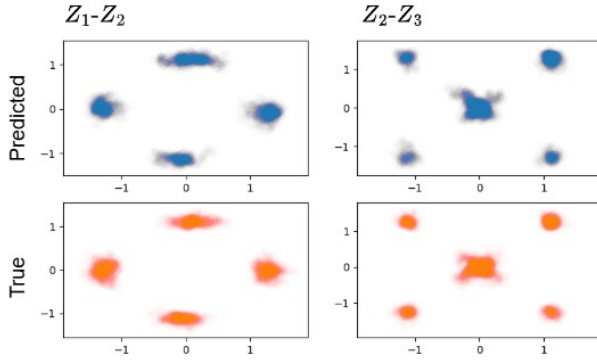
(a) Batch size: 1



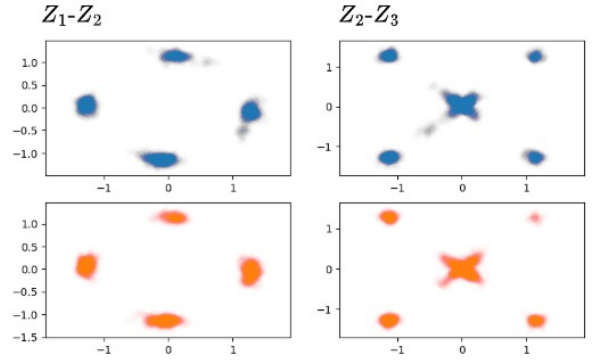
(b) Batch size: 64



(c) Batch size: 1



(d) Batch size: 64



Supplementary Figure 6: Predicted versus true ICA with SGLD trajectories and invariant distributions under varying batch sizes. Time series plots (a, b) and scatter plots (c, d) demonstrate the consistency between the predicted and true latent trajectories across four ICA components. Panels (a) and (b) depict the temporal evolution over 2000 steps with batch sizes (1) and (64), respectively. Panels (c) and (d) compare the corresponding invariant distributions, showing scatter plots of (Z_1, Z_2) and (Z_2, Z_3) .

and a potential function $V_b(\mathbf{Z})$ such that the drift term can be decomposed as:

$$-\nabla L(\mathbf{Z}) = -[M_b(\mathbf{Z}) + W_b(\mathbf{Z})] \nabla V_b(\mathbf{Z}) + \nabla \cdot M_b(\mathbf{Z}) + \nabla \cdot W_b(\mathbf{Z}),$$

where $M_b(\mathbf{Z}) = \frac{\eta}{2}\Sigma_b(\mathbf{Z}) + 1$. Denoting by p_b the stationary density of SGLD with batch size b , and by p_∞ that of the full-batch SGLD, we obtain:

$$\dot{S}_{\text{tot}} = \int_{\mathbb{R}^D} f_{\text{irr}}^\top M^{-1} f_{\text{irr}} \rho(\mathbf{Z}) d\mathbf{Z} = \mathbb{E}_{\mathbf{Z} \sim p_b} \left[f_{\text{irr}}(\mathbf{Z})^\top \left(\frac{\eta}{2} \Sigma_b(\mathbf{Z}) + 1 \right)^{-1} f_{\text{irr}}(\mathbf{Z}) \right], \quad (2)$$

and

$$\nabla \ln p_b = -\nabla V_b, \quad \nabla \ln p_\infty = -\nabla L.$$

Next we introduce the Relative Fisher Information (RFI) between distributions with densities p and q . This asymmetric divergence measures the discrepancy between their score functions:

$$I(p, q) = \mathbb{E}_{\mathbf{Z} \sim p} \left[\|\nabla \ln p(\mathbf{Z}) - \nabla \ln q(\mathbf{Z})\|^2 \right].$$

RFI provides an upper bound for the Wasserstein distance and the KL divergence according to HWI inequality [3] under the Bakry-Émery condition, with applications in information geometry and statistical physics [4, 2]. For SGLD, the RFI is given by

$$\begin{aligned} I(p_b, p_\infty) &= \mathbb{E}_{\mathbf{Z} \sim p_b} \left[\|\nabla \ln p_b(\mathbf{Z}) - \nabla \ln p_\infty(\mathbf{Z})\|^2 \right] \\ &= \mathbb{E}_{\mathbf{Z} \sim p_b} \left[\|\nabla V_b(\mathbf{Z}) - \nabla L(\mathbf{Z})\|^2 \right]. \end{aligned}$$

Substituting Eq. (1), we obtain:

$$\begin{aligned} \nabla L(\mathbf{Z}) - \nabla V_b(\mathbf{Z}) &= [M_b(\mathbf{Z}) + W_b(\mathbf{Z})] \nabla V_b(\mathbf{Z}) - (\nabla \cdot M_b(\mathbf{Z}) + \nabla \cdot W_b(\mathbf{Z})) - \nabla V_b(\mathbf{Z}) \\ &= \left[\frac{\eta}{2} \Sigma_b(\mathbf{Z}) \nabla V_b(\mathbf{Z}) - \nabla \cdot \left(\frac{\eta}{2} \Sigma_b(\mathbf{Z}) \right) \right] + f_{\text{irr}}(\mathbf{Z}) \\ &= \frac{\eta}{2} [\Sigma_b(\mathbf{Z}) \nabla V_b(\mathbf{Z}) + \nabla \cdot \Sigma_b(\mathbf{Z})] + f_{\text{irr}}(\mathbf{Z}). \end{aligned}$$

Thus,

$$I(p_b, p_\infty) = \mathbb{E}_{\mathbf{Z} \sim p_b} \left[\left\| \frac{\eta}{2} [\Sigma_b(\mathbf{Z}) \nabla V_b(\mathbf{Z}) + \nabla \cdot \Sigma_b(\mathbf{Z})] + f_{\text{irr}}(\mathbf{Z}) \right\|^2 \right]. \quad (3)$$

The difference between Eq. (2) and Eq. (3) is attributed to the Σ_b term, reflecting the effect of minibatch-induced stochasticity and diminishing as the batch size increases. This observation provides a theoretical justification for the empirical finding that both the sampling error and the global EPR consistently decrease with larger batch sizes. In addition, since all quantities involved (V_b , W_b , M_b , and consequently f_{irr} and Σ_b) are explicitly parameterized and estimable within our framework UIDD, Eq. (3) can be computed directly from data and thus serves as a principled scalar indicator of sampling error in SGLD.

F Supplementary Information for Synthetic Examples

F.1 Linear Case

We describe the data generation procedure. The dataset is created by simulating a target linear SDE of the form:

$$\dot{\mathbf{Z}}_t = -(M + W)S\mathbf{Z}_t + \sqrt{2M}\dot{\mathbf{B}}_t,$$

where M and S are symmetric positive-definite matrices, and W is antisymmetric with $W = \lambda W_0$. This formulation is designed to capture the combined effects of symmetric dissipation, antisymmetric coupling, and stochastic forcing.

The coefficient matrices M , W , and S are obtained by first sampling Gaussian random matrices and normalizing them. Subsequently, symmetry constraints are imposed: M and S are symmetrized and scaled to ensure positive definiteness, whereas W is rendered strictly antisymmetric. As a result, the generated system satisfies the intended structural properties by construction.

Once the matrices are fixed, the SDE is integrated over the time interval $[0, 1]$ using a constant time step of 0.01. Here, initial conditions are drawn from a Gaussian distribution with zero mean and standard deviation 2, and each trajectory is driven by an independent Brownian motion. In total, 10^4 independent trajectories are simulated, of which 9000 are allocated to the training set and 1000 are reserved for the validation set.

The potential module (V) consists of two fully connected layers with 64 and 32 hidden units, respectively, activated by the ReCU function. The anti-symmetry module (W) includes two hidden layers with 32 and 16 units, also using the ReCU activation. The entire model is trained using the Adam optimizer for 5000 epochs with a batch size of 50,000.

Relative error of structural recovery is measured by

$$e_f = \left(\frac{\sum_x \|g_\theta(x) - g(x)\|^2}{\sum_x \|g(x)\|^2} \right)^{1/2},$$

where the sum is taken over a set of $N = 10^3$ states sampled independently from the standard normal distribution.

As a non-physics-based reference, an unstructured SDENet trained on the same data attains a drift relative error of $9.56 \times 10^{-3} \pm 3.7 \times 10^{-4}$ (mean $\pm$ std over random seeds), comparable to UIDD in fitting the dynamics while not recovering the underlying potential or reversible/irreversible components.

F.2 Nonlinear Case

Set-up. We use a two-dimensional nonequilibrium diffusion to test whether a learned SDE can recover not only the observed transition law but also the thermodynamic decomposition that generates it. The target dynamics are

$$dX_t = -[M + W]\nabla V(X_t) dt + \sqrt{2T} M^{1/2} dB_t, \quad (4)$$

where the temperature is $T = 1$,

$$M = \epsilon I, \quad W = \epsilon \kappa \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \epsilon = 0.01, \quad \kappa = 0.8,$$

and the double-well potential is

$$V(x_1, x_2) = \frac{\frac{1}{4}(x_1^2 - 1)^2 + (3.5 + \sin x_1)x_2^2}{\epsilon}.$$

The potential induces a bimodal invariant distribution, while the antisymmetric component W generates non-gradient circulation. Since M and W are constant matrices, the required divergence-free conditions are satisfied, and the target SDE Eq. (4) is covered by the UIDD parameterization. Thus, V coincides with the negative log-density of the invariant distribution up to an additive constant,

while $M\nabla V$ and $W\nabla V$ coincide with the time-reversible and time-irreversible components of the drift. The same SDE also satisfies the structural assumptions of OnsagerNet, but its representation within OnsagerNet is not identifiable: the same drift admits multiple decompositions into V , M , and W . Accurate drift fitting is therefore not enough for this example. A successful model must also recover the correct potential, reversible component, and irreversible component.

Data. Training trajectories are generated from the exact target SDE. Initial conditions are sampled uniformly from $[-2.5, 2.5] \times [-1.5, 1.5]$. The SDE is integrated with internal time step 0.001, and observations are recorded every $\Delta t = 0.01$ on $[0, 0.5]$, giving 51 observations per trajectory. The dataset contains 9×10^3 training trajectories and 1×10^3 test trajectories. During training, each trajectory is reshaped into $\lfloor 51/2 \rfloor = 25$ non-overlapping adjacent two-point segments, yielding 2.25×10^5 one-step pairs. All models are trained using the Euler–Maruyama maximum-likelihood objective on these pairs.

Baseline. We compare UIDD, OnsagerNet, and an unstructured SDENet baseline. To keep the comparison focused on the drift parameterization, the diffusion is parameterized identically in all three models as a trainable but state-independent matrix in lower-triangular Cholesky form. The drift parameterizations differ. SDENet uses a generic feed-forward network with two hidden layers of 128 units each and tanh activation for the total drift, providing a flexible transition-density baseline without any thermodynamic decomposition. OnsagerNet uses a two-hidden-layer potential network V_θ with hidden widths 64 and 32, together with two matrix-valued networks for M_θ and W_θ each with hidden widths 32 and 16. UIDD reuses the same potential network as OnsagerNet and adds a Hamiltonian network with hidden widths 32 and 16 that determines the time-irreversible drift, while its time-reversible part is fixed by the shared diffusion matrix via the fluctuation–dissipation relation. All structured component networks of OnsagerNet and UIDD use the rectified cubic unit (RECU) activation. Unless otherwise specified, all models are trained for 3×10^3 epochs with learning rate 10^{-3} and batch size 1×10^5 over the one-step pairs, repeated over 5 independent random seeds; these defaults define the configuration whose results appear in the main comparison and in the regularization study, and they are systematically varied only in the UIDD hyperparameter sensitivity study reported at the end of this section.

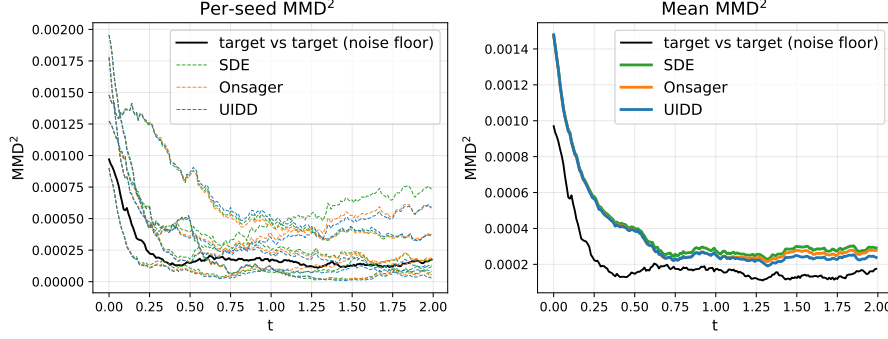
Evaluation protocol. For each seed, we compare $N = 10^3$ sample paths from the learned model with $N = 10^3$ sample paths from the target SDE. Their initial states are generated independently and sampled uniformly from $[-2, 2] \times [-1, 1]$. All sample paths are evolved over $[0, 2.0]$, four times longer than the training horizon. At each observation time t_k , distributional accuracy is measured by the Gaussian-kernel empirical squared MMD

$$\text{MMD}^2(t_k) = \frac{1}{N^2} \sum_{i,j=1}^N k(x_i(t_k), x_j(t_k)) + \frac{1}{N^2} \sum_{i,j=1}^N k(\hat{x}_i(t_k), \hat{x}_j(t_k)) - \frac{2}{N^2} \sum_{i,j=1}^N k(x_i(t_k), \hat{x}_j(t_k)),$$

where

$$k(u, v) = \exp\left(-\frac{\|u - v\|^2}{\sigma^2}\right),$$

and $\{x_i(t_k)\}_{i=1}^N$ denotes the reference trajectories generated by the target SDE, whereas $\{\hat{x}_i(t_k)\}_{i=1}^N$ denotes the corresponding predictions generated by the learned model. A single bandwidth, shared across all time points, is fixed by the median heuristic on a pooled set of up to 4×10^3 states



Supplementary Figure 7: Prediction accuracy for the nonlinear SDE experiment, measured by Gaussian-kernel MMD^2 over time. The target-versus-target curve represents the finite-sample noise floor.

sampled from an independent target ensemble. A target-versus-target curve obtained by drawing two independent target ensembles is included as the finite-sample noise floor.

Relative error of structural recovery is measured on target-sampled states by

$$e_f = \left(\frac{\sum_x \|g_\theta(x) - g(x)\|^2}{\sum_x \|g(x)\|^2} \right)^{1/2},$$

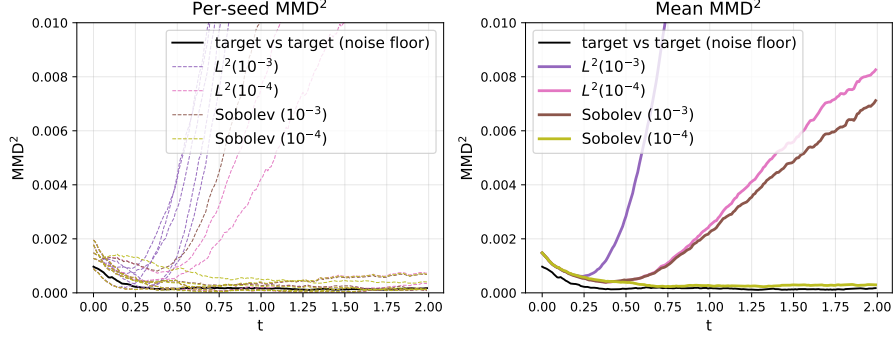
where the sum is taken over a fixed set of $N = 2 \times 10^3$ states drawn uniformly at random from a separate target ensemble (simulated under the same setting as the MMD evaluation above). The same x -set is shared across all evaluated models. SDENet is evaluated only on the total drift as it does not provide a thermodynamic decomposition. OnsagerNet and UIDD are also evaluated on ∇V , $M\nabla V$, and $W\nabla V$, where $M\nabla V$ corresponds to the time-reversible drift and $W\nabla V$ to the time-irreversible drift.

Method	Drift	∇V	$M\nabla V$	$W\nabla V$
SDENet	$4.508 \times 10^{-2} \pm 6.71 \times 10^{-4}$	N/A	N/A	N/A
OnsagerNet	$4.660 \times 10^{-2} \pm 3.09 \times 10^{-3}$	$9.917 \times 10^{-1} \pm 3.38 \times 10^{-3}$	$4.828 \times 10^{-1} \pm 3.30 \times 10^{-1}$	$6.116 \times 10^{-1} \pm 4.02 \times 10^{-1}$
UIDD	$4.680 \times 10^{-2} \pm 2.19 \times 10^{-3}$	$4.238 \times 10^{-2} \pm 2.16 \times 10^{-3}$	$3.941 \times 10^{-2} \pm 3.04 \times 10^{-3}$	$5.195 \times 10^{-2} \pm 2.29 \times 10^{-3}$

Supplementary Table 3: Relative structural errors for the three models, reported as mean $\pm$ standard deviation over 5 seeds. Bold numerical entries mark the smallest mean error in each column. SDENet does not provide a thermodynamic decomposition, so the corresponding component errors are unavailable.

Results. Supplementary Figure 7 reports the distributional accuracy. All three models achieve comparable trajectory matching: their MMD^2 curves rapidly approach the target-versus-target noise floor, and the differences among the learned models are small compared with the overall decrease in MMD^2 after the initial transient. UIDD has slightly better average and final MMD^2 values, but its main advantage is structural recovery rather than trajectory-level matching alone.

Supplementary Table 3 shows that structural recovery separates UIDD from the baselines. All models learn a similarly accurate aggregate drift, which indicates that the data contain enough information to fit the observed short-time dynamics. OnsagerNet, however, does not identify the correct latent decomposition: it can compensate for errors in V_θ , M_θ , and W_θ while keeping a reasonable total drift. UIDD avoids this ambiguity through its identifiable parameterization, which



Supplementary Figure 8: Prediction accuracy for regularized OnsagerNet. We test both L^2 and Sobolev penalties. The number in parentheses denotes the regularization weight.

prevents the model from redistributing the same drift among non-identifiable components. It therefore retains the same level of predictive accuracy as the baselines while recovering each physical component of the target dynamics.

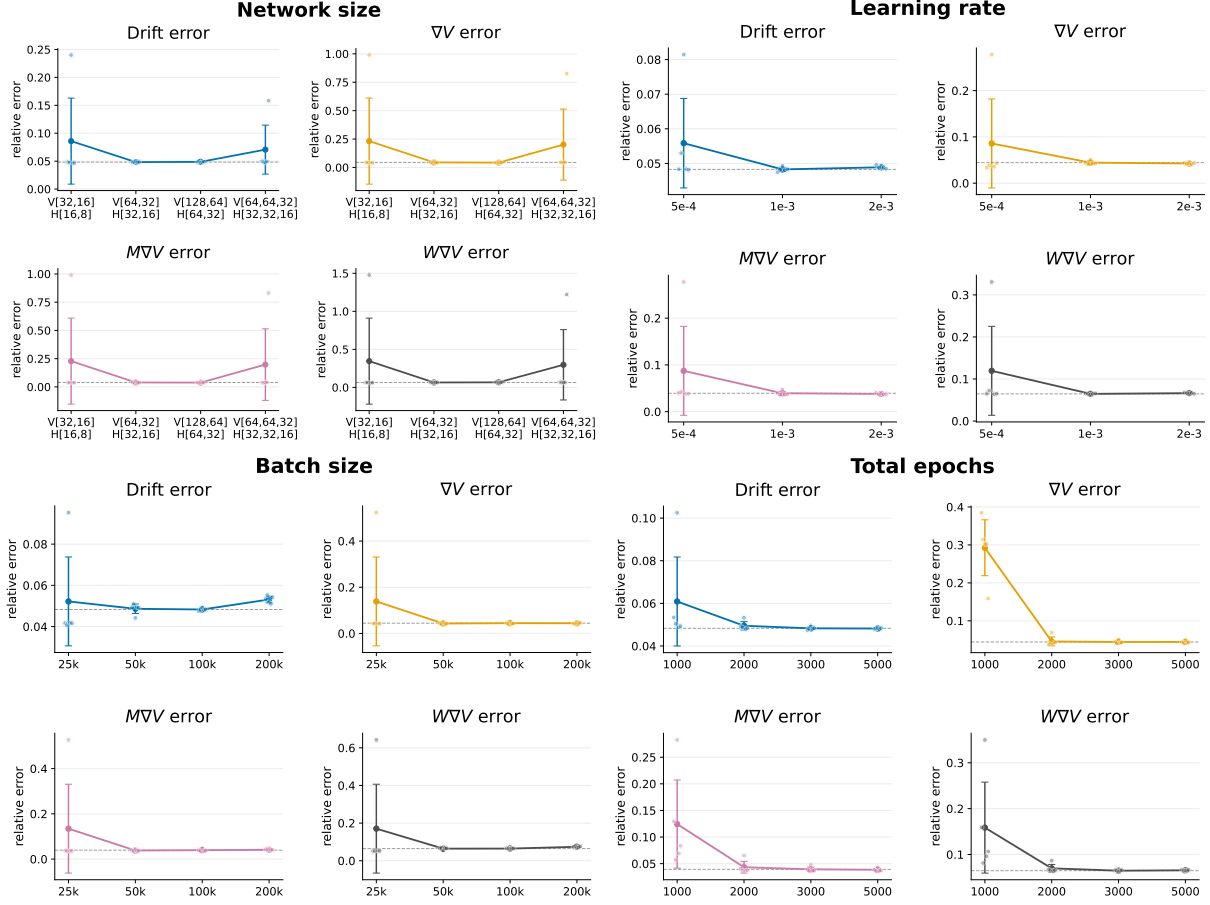
Regularizer	Drift	∇V	$M\nabla V$	$W\nabla V$
None	$4.660 \times 10^{-2} \pm 3.09 \times 10^{-3}$	$9.917 \times 10^{-1} \pm 3.38 \times 10^{-3}$	$4.828 \times 10^{-1} \pm 3.30 \times 10^{-1}$	$6.116 \times 10^{-1} \pm 4.02 \times 10^{-1}$
$L^2 (10^{-3})$	$1.646 \times 10^{-1} \pm 1.66 \times 10^{-3}$	$9.997 \times 10^{-1} \pm 6.78 \times 10^{-5}$	$6.653 \times 10^{-1} \pm 7.47 \times 10^{-2}$	$8.424 \times 10^{-1} \pm 8.87 \times 10^{-2}$
$L^2 (10^{-4})$	$7.260 \times 10^{-2} \pm 3.21 \times 10^{-2}$	$9.986 \times 10^{-1} \pm 3.26 \times 10^{-4}$	$3.739 \times 10^{-1} \pm 1.27 \times 10^{-1}$	$4.867 \times 10^{-1} \pm 1.64 \times 10^{-1}$
Sobolev (10^{-3})	$6.539 \times 10^{-2} \pm 3.72 \times 10^{-2}$	$9.993 \times 10^{-1} \pm 6.77 \times 10^{-5}$	$4.593 \times 10^{-1} \pm 2.65 \times 10^{-1}$	$5.903 \times 10^{-1} \pm 3.30 \times 10^{-1}$
Sobolev (10^{-4})	$4.662 \times 10^{-2} \pm 1.47 \times 10^{-3}$	$9.981 \times 10^{-1} \pm 1.31 \times 10^{-4}$	$3.682 \times 10^{-1} \pm 1.03 \times 10^{-1}$	$4.790 \times 10^{-1} \pm 1.27 \times 10^{-1}$

Supplementary Table 4: Relative structural errors for OnsagerNet trained with an additional regularization penalty, reported as mean $\pm$ standard deviation over 5 seeds. We test both L^2 and Sobolev penalties. The number in parentheses denotes the regularization weight. Bold entries mark the smallest mean error in each column.

Regularizing OnsagerNet. To test whether standard regularization can remove the ambiguity in OnsagerNet, we also train OnsagerNet with two common penalties applied to the potential network V_θ : an L^2 penalty $\lambda \sum_i \theta_i^2$ on its parameters and a Sobolev-style penalty $\lambda \|\nabla^2 V_\theta(x)\|_F^2$ on the Frobenius norm of its Hessian, evaluated at training samples. For each penalty, we use weights 10^{-3} and 10^{-4} , together with the same 5 random seeds and the same evaluation protocol as above.

Supplementary Figure 8 and Supplementary Table 4 show that standard regularization can influence trajectory matching and reduce the seed-to-seed variance of the ∇V error, but it does not remove the structural non-identifiability of OnsagerNet. Among the regularized variants, OnsagerNet with a Sobolev penalty of weight 10^{-4} gives the best overall performance. Nevertheless, the mean error in ∇V remains close to one, and the errors in both $M\nabla V$ and $W\nabla V$ remain substantial. These results indicate that the structural failure of OnsagerNet reported in Supplementary Table 3 is not merely due to the absence of a standard regularizer; rather, explicit identifiability is necessary for the stable recovery of interpretable physical structures.

UIDD hyperparameter sensitivity. To check whether the structural recovery reported in Supplementary Table 3 depends on a particular hyperparameter choice, we conduct a one-factor-at-a-time sensitivity study around the default UIDD configuration introduced above. We vary the network size using three alternatives: a smaller variant with potential/Hamiltonian widths (32, 16)/(16, 8), a wider variant (128, 64)/(64, 32), and a deeper variant (64, 64, 32)/(32, 32, 16). We



Supplementary Figure 9: UIDD hyperparameter sensitivity for network width, learning rate, batch size, and number of training epochs. Markers show seed averages, error bars show seed-to-seed standard deviations, and lightly jittered points show individual seeds.

also separately sweep the learning rate, batch size, and number of training epochs. The default configuration is included in every sweep so that sensitivity to each hyperparameter can be compared directly. For the epoch sweep, a single 5×10^3 -epoch run is checkpointed and evaluated at 1×10^3 , 2×10^3 , 3×10^3 , and 5×10^3 epochs.

Supplementary Figure 9 shows that the default configuration is already close to the best observed structural recovery for every quantity. Increasing the network width gives slightly lower ∇V and $M\nabla V$ errors but does not materially improve the total drift or the time-irreversible component. The deeper and smaller variants each produce one outlier seed with much larger component errors, which increases both the mean and the variance. The baseline width is therefore sufficient for this two-dimensional example, and additional depth does not provide a clear benefit.

The learning-rate and batch-size sweeps show the same qualitative pattern. Doubling the learning rate to 2×10^{-3} keeps training stable and yields component errors comparable to the default, whereas halving it to 5×10^{-4} is less reliable within the fixed 3×10^3 -epoch budget, with one seed failing to reach the same structural accuracy. Batch size 5×10^4 is essentially indistinguishable from the default; 2×10^5 slightly worsens the drift and $W\nabla V$ errors; and 2.5×10^4 performs well for most seeds but is affected by one unstable seed. The epoch sweep shows that structural recovery is largely completed between 1×10^3 and 2×10^3 epochs. At 2×10^3 epochs, the errors are already

close to those obtained with the default 3×10^3 -epoch setting, and increasing the training length to 5×10^3 epochs leads only to minor changes. These sweeps indicate that the structural-recovery conclusions are stable under the tested changes in network width, learning rate, batch size, and number of training epochs.

F.3 A twenty-dimensional nonlinear SDE

We also test the similar setup on a coupled twenty-dimensional variant. The changes to the target SDE are the potential and the antisymmetric matrix:

$$V(x) = \frac{\frac{1}{4}(x_1^2 - 1)^2 + (3.5 + \sin x_1) \sum_{i=2}^{20} x_i^2}{\epsilon}, \quad W = \epsilon \kappa J,$$

where $\epsilon = 0.01$, $\kappa = 0.8$, and $J_{i,i+1} = -1$, $J_{i+1,i} = 1$ for $i = 1, \dots, 19$, with all other entries zero. Thus x_1 is the single double-well coordinate and it couples to the other nineteen coordinates through the common quadratic factor.

Initial conditions are randomly sampled from $[-2.5, 2.5] \times [-1.5, 1.5]^{19}$. We use 9×10^4 training trajectories and 10^4 test trajectories, giving 2.25×10^6 one-step training pairs after the same two-point reshaping as above. We train UIDD and OnsagerNet over the same five seeds. The computational cost remains moderate even in this high-dimensional setting. On the same computational platform, a single run for the 20-dimensional example takes approximately one hour, whereas a run for the two-dimensional examples takes approximately twenty minutes.

Method	Drift	∇V	$M\nabla V$	$W\nabla V$
OnsagerNet	$7.110 \times 10^{-2} \pm 8.56 \times 10^{-3}$	$9.999 \times 10^{-1} \pm 1.71 \times 10^{-5}$	$3.361 \times 10^{-1} \pm 6.76 \times 10^{-2}$	$3.093 \times 10^{-1} \pm 6.06 \times 10^{-2}$
UIDD	$5.586 \times 10^{-2} \pm 9.01 \times 10^{-4}$	$6.049 \times 10^{-2} \pm 7.30 \times 10^{-4}$	$2.962 \times 10^{-2} \pm 3.52 \times 10^{-4}$	$7.074 \times 10^{-2} \pm 1.31 \times 10^{-3}$

Supplementary Table 5: Relative structural errors for the twenty-dimensional nonlinear SDE, reported as mean $\pm$ standard deviation over 5 seeds. Bold entries mark the smaller mean error in each column.

Supplementary Table 5 shows that UIDD accurately recovers the potential gradient, reversible drift, and irreversible drift in the twenty-dimensional nonlinear system. OnsagerNet attains a comparatively reasonable approximation of the total drift, but its errors in the individual components remain substantially larger. These results indicate that the conclusion drawn from the two-dimensional example persists in higher dimensions.

References

- [1] Q. Li, C. Tai, et al. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47, 2019.
- [2] N. Mukherjee and A. K. Roy. Relative fisher information in some central potentials. *Annals of Physics*, 398:190–202, 2018.
- [3] F. Otto and C. Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173(2):361–400, 2000.
- [4] C. Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.